

Astrostatistics School: requirements.

By S. Andreon & R. Trotta

Participants to the school are requested to install a few software packages and to write some reading/plotting routines prior to the beginning of the school. Failing to do so will significantly impair the learning outcomes.

Attendees should:

1. have their own computer (with an internet connection), and be acquainted with it;
2. have installed JAGS¹ which in turn may demand the installation of some additional libraries;
3. be able to make plots and simple data manipulation using their preferred environment. In particular, attendees should have already written routines for:

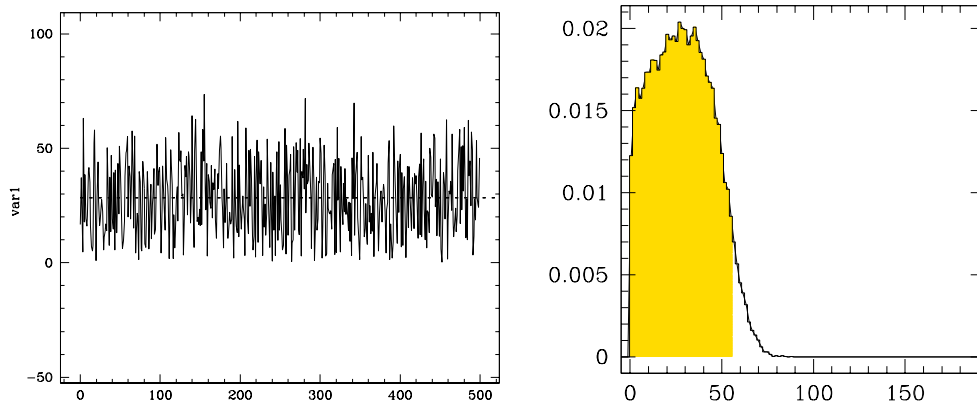


Figure 1: Left panel: Trace plot. Right panel: Marginal distribution (histogram)

- properly reading files in the CODA format: CODAindex.txt² describes the content of the CODAchain1.txt³ file by listing the variable names, where they start and where they end. For example (inspect CODAindex.txt) the variable s starts at line 1 and ends at 50,000 (and it is on the 2nd column) of CODAchain1.txt. The reading routine should work for any number of variables (e.g. 10) and of samples (e.g. 30,000).
- compute mean and standard deviation (check that s has mean 28.5 and standard deviation 16.5)

¹<https://sourceforge.net/projects/mcmc-jags/>

²http://www.brera.mi.astro.it/~andreon/corso_metodi_bayesiani/CODAindex.txt

³http://www.brera.mi.astro.it/~andreon/corso_metodi_bayesiani/CODAchain1.txt

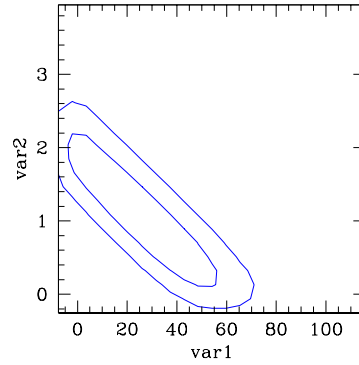


Figure 2: Contours. In this plot, contours are imprecisely determined close to $var1 = 0$ e $var2 = 0$ boundaries, and not corrected for smoothing effects. You are not asked to do better (but nothing precludes you from doing it).

- compute the shortest interval including $x\%$ of the samples (check that the 95% interval of s is $[0,56]$). To compute it, you may, for example, start from the peak of the pdf and move down until they get $x\%$ of the samples.
 - produce a trace plot, i.e., a plot that gives the variable value as a function of its rank (or step in the chain), as in Fig. 1. This plotting routine should work also if CODAindex.txt contains, say, 10 variables.
 - produce normalized histograms, as in Fig. 1, right panel (note that the integral must be unity and independently of bin size).
 - draw contours. The routine should work for non-elliptical contours, for example when one has two separate “islands”. The contours should include about 68% and about 95% of the samplings. A small margin of error is allowed (i.e. 70% in place of 68% is fine). It is instead not allowed to draw contours at pre-defined thresholds (e.g, taking the peak value and dividing by a “magic number”). Check your contours against those in Fig. 2 with the sampling in CODAchain1.txt. The latter contours are somewhat approximated (and nothing better than this is required!).
4. Advanced students may *optionally* install `PyMultinest`⁴ and, if they use Python, the supplied Jupyter notebook⁵. The latter provides some of the functionalities indicated above:
- compute mean and standard deviation: use the Python function `numpy.mean` and `numpy.std` on the appropriate array dimension.
 - produce (un)normalized histograms: use the function `hist(S2.trace(variable_name)[:])`
 - draw contours: use the function `plot_MCMC_trace`.

`PyMultinest` and Jupyter notebook are useful, but not required.

⁴PyMultinest requires a working installation of the MultiNest software on your computer, to which it interfaces. This in turn requires C and Fortran compilers, and a number of scientific libraries to link to, see <http://johannesbuchner.github.io/PyMultiNest/install.html#installing-the-python-module> for details.

⁵http://www.brera.mi.astro.it/~andreon/corso_metodi_bayesiani/LinearModelJupyterPyMC-Exercices.ipynb

Checking JAGS installation

In order to check having properly installed JAGS, first, save the file below as model.bug and make you sure that it only contain ASCII

```
# Bayesian Methods for the Physical Science. Learning from
# Examples in Astronomy and Physics. By S. Andreon and B. Weaver.
model {
  for (i in 1:length(nrec)) {
    nrec[i] ~ dbin(eff[i],ninj[i])
    nrec.rep[i] ~ dbin(eff[i],ninj[i])
    eff[i] <- A + (B-A)*phi((E[i]-mu)/sigma)
  }
  A~dunif(0,1)
  B~dunif(0,1)
  mu~dunif(0,100)
  sigma~dunif(0,100)
}
```

Second, save the file below as model.cmd and check that it only contain ASCII

```
model in model.bug
data in data.dat.R
compile,nchains(1)
initialize
update 3000
monitor set A, thin(10)
monitor set B, thin(10)
monitor set mu, thin(10)
update 100000
coda *
data to testata
samplers to testsamplers
exit
```

Now, using the data listed at this URL⁶ (save the file as data.dat.R), run JAGS redirecting the standard input from the file model.cmd. Under linux the command to execute is

```
jags < model.cmd
```

provided that the jags executable is in the path. If a file CODAchain1.txt is produced, then JAGS has been properly installed.

⁶<http://www.brera.mi.astro.it/%7Eandreon/BayesianMethodsForThePhysicalSciences/data8.2.dat.R>